［利用相談室便り］

自己紹介と研究室の紹介

佐藤 義永
東北大学大学院情報科学研究科情報基礎科学専攻 博士課程後期 2 年

今年度より,東北大学サイバーサイエンスセンターにおいて火曜日の 10:00～12:00 に利用相談を担当させていただくことになりました佐藤義永と申します．スーパーコンピュータや並列コンピュータの操作方法，プログラムの高速化について担当しています．

私は東北大学大学院情報科学研究科情報基礎科学専攻の超高速情報処理論講座という研究室に所属しております．2010 年 7 月現在，学部生 12 人，院生 17 人，スタッフ 8 人(うち秘書 2 名)が研究室の構成員です。私たちの研究室で取り組んでいる研究は，高性能かつ低消費電力なマイクロプロセッサに関する研究や，グラフィックス処理用ハードウェアを用いた高性能計算に関する研究，P2P やグリッド等のネットワークを利用した大規模計算環境に関する研究，そしてサイバーサイエンスセンターで利用されるベクトル型スーパーコンピュータのさらなる性能向上を目指した基礎研究といった計算機アーキテクチャや大規模計算環境に関する研究を行っています．また，高機能文書認識システムや画像認識，ユビキタスコンピューティングなどのアプリケーションに関する研究にも取り組んでいます．これらの研究の中から本稿では，ベクトル型スーパーコンピュータに関する研究を紹介させていただきます．

大規模かつ高精度な科学技術シミュレーションを実現するには高い演算性能を有する計算機が求められますが，それだけでは必ずしも期待通りの性能は得られません．科学技術シミュレーションでは，メインメモリからプロセッサへ計算に要する膨大な量のデータ転送が行われます．もしもデータ転送性能が演算性能に見合っていなければ，データの転送に要する時間がプログラムの実行時間の大半を占めることとなり，プロセッサの演算性能が活かせなくなってしまいます．そこで，高いデータ転送性能を有するベクトルプロセッサに着目し，ベクトルプロセッサにより構成されるベクトル型スーパーコンピュータのさらなる高性能化を目指しています．

ベクトルプロセッサは，メインメモリを独立してデータ転送可能な複数のメモリバンクに分割するインターリーブ方式のメモリアーキテクチャを採用しています．これにより，メインメモリからプロセッサへとデータを並列に転送することができ，高いデータ転送性能を実現することで演算性能との差を埋めています.このような特徴から，大規模な科学技術計算において，ベクトル型スーパーコンピュータは高い実行効率を実現してきました．

しかし近年では，プロセッサに実装可能な入出力ピン数の物理的限界から，データ転送性能の向上が今まで以上に困難となってきています．一方でプロセッサの演算性能は半導体チップの微細化による集積度向上により，メモリバンド幅以上に向上しています．そこで，演算性能（GFlop/s）とデータ転送性能（GB/s）の関係の指標としてピーク演算性能とメモリバンド幅の比（B/F）に着目すると，B/F の減少は実行効率の低下を招くことが過去の研究から示されています[1]．したがって，データ転送性能の向上による B/F の改善が次世代のスーパーコンピュータ開発における重要な課題となっています．

私の研究では，ベクトル型スーパーコンピュータを高効率で利用するためのプログラム最適化手法に関する研究を行っています．近年のベクトルプロセッサにはデータ転送性能を補うためにキャッシュメモリ(ベクトルキャッシュ)の搭載が検討されています．キャッシュメモリとは，プロセッサの内部に設けられる高速なメモリで，メインメモリから読み出したデータをキャッシュ内部に保存し，次に参照される際にデータを高速に転送する機構です．アプリケーションには，

一度参照したデータは再び利用されやすいという時間的局所性と，参照するデータの付近にあるデータも参照されやすいという空間的局所性という性質があります．これらの性質から，キャッシュを用いてデータを高速に転送することでデータ転送性能の改善を図ることができます．

しかし，ベクトルキャッシュは高速である代わりに，データを保存可能な容量がメインメモリに比べて非常に小さいという欠点があります．現在のプロセッサでは数 MB 程度の容量が一般的です．したがって，大規模科学技術計算では，扱うデータが大規模であることからすべてのデータをキャッシュに格納することは不可能です．このことから，ベクトルキャッシュを用いて性能向上を実現するには，キャッシュを活用するためのプログラム最適化手法が必要になります．

従来のベクトルプロセッサ向けのプログラム最適化では，ベクトル長を長くすることで，ベクトル命令で一度に扱うデータ量を増やし，ループ分岐のオーバーヘッドを隠ぺいすることにより高速化を図ってきました．一方，キャッシュの利用を考えた場合，ベクトル長は短い方が時間的局所性は大きくなる場合があります．したがって，ベクトル長を短くするようにループを分割するキャッシュブロッキングと呼ばれる最適化手法が有効になります．

このように，従来のベクトルプロセッサ向けの最適化とベクトルキャッシュ向けの最適化では効果が相反する場合があります．私の研究では，複数の最適化の活用を考慮したベクトルキャッシュを有するベクトルプロセッサのための最適化戦略の確立を目指しています．これを実現するために，ベクトルキャッシュを加えたベクトルプロセッサを想定した性能モデルを構築し，性能モデルに基づいて効率的にプログラムを最適化する戦略を現在検討しています．

前述の通り，ベクトルプロセッサで利用される科学技術アプリケーションでは，プロセッサとメインメモリ間におけるデータ転送時間が実行時間の大部分を占めています．そのため，システムの実効性能はプロセッサとメインメモリ間のデータ転送性能に大きく左右されることになります．このことから，ベクトルプロセッサの性能モデルはプロセッサの演算性能だけではなくデータ転送性能も考慮する必要があります．そこで，ルーフラインモデルをベクトルプロセッサの性能モデルとして採用します．ルーフラインモデルの重要な点は，性能指標として演算密度を用いる点にあります．演算密度とは，アプリケーションに含まれる演算量と，演算に必要なデータ転送量(プロセッサ-メインメモリ間)の比であり，これを利用することによりシステムとアプリケーションの双方の特徴から性能を考慮することができます．なお，演算密度の単位として，Flops per Byte が一般的に用いられます．結果として，アプリケーション毎に性能のボトルネックの解析が可能となり，アプリケーションとシステムの特徴を踏まえたプログラム最適化の指針を導くことができます．

図1 にメモリバンド幅が2B/F時のルーフラインモデルを示します．この性能モデルより読み取れることは，プログラムの演算密度が1/2 以上であれば，実行効率の上限が100%に達し，データ転送性能はボトルネックにならないため，プロセッサの性能を活かすようにループ内の演算数を増やすループアンローリング等の最適化が求められます．一方で，演算密度が1/2以下でなおかつアプリケーションの実行効率がデータ転送性能の上限に達している場合は，データ転送性能がボトルネックとなっているため，ベクトルキャッシュを活用する必要があります．したがってこの場合はキャッシュブロッキング等の最適化が有効であると考えられます．

性能モデルに基づいてボトルネックの解析を行い，ボトルネックを解消するようにプログラム最適化を施した際の性能評価を最後に示します．本評価を通じてプログラム最適化の有効性を検証します．また，プログラム最適化の有無による消費エネルギへの影響も評価し，消費エネルギの面からもプログラム最適化の有効性を検証します．評価には，地震解析，電磁場解析の分野から東北大学サイバーサイエンスセンターで実際に利用されているアプリケーションを用い，流体解析にはポアッソン方程式をヤコビの反復法で解く際の主要ループで構成される姫野ベンチマーク[4]を用います．評価アプリケーションを表1にまとめて示します．

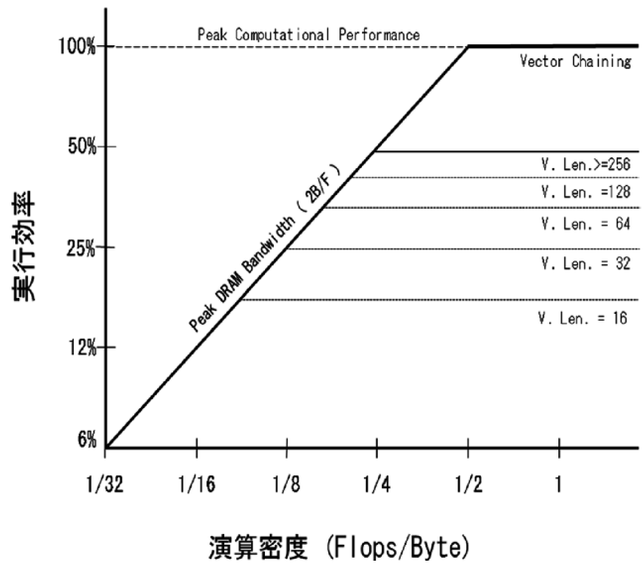

図 1 ベクトルプロセッサにおけるルーフラインモデル

表 1　評価アプリケーション一覧

評価アプリケーション	計算手法	問題サイズ	ベクトル化率	ベクトル長
地震解析 [3]	Friction Law	2047×2047×257	99.5 %	256
流体解析 [4]	3-D Jacobi	512×512×256	99.5 %	256
地雷探索 [5]	FDTD	1500×1500×50	99.7 %	250

評価結果を図2に示します．地震解析による評価では，プログラム最適化により約3倍に性能向上しています．地震解析のアプリケーションはループ内の演算数が少なく，データ転送がプログラムの実行時間の大半を占めています．さらに，ルーフラインモデルの解析により演算密度に対してデータ転送性能はボトルネックになっていないことが分かります．そこで，ループ内の演算数を増加させるためにループアンローリングを施すことによって大幅な性能向上が得られます．

流体解析や地雷探索のプログラムでは，ルーフラインモデルにより，ともに演算密度が低くデータ転送性能の上限に達している，すなわちデータ転送性能がボトルネックであることがわかります．そこで，キャッシュブロッキングによる最適化を施すことにより，ベクトルキャッシュが活用でき，平均で5%の性能向上が得られます．したがって，性能モデルに基づくボトルネック解析により，ベクトルキャッシュを活用したプログラム最適化が適用可能であることがわかります．

次に，評価アプリケーション実行時の消費エネルギの評価結果を図3に示します．プログラム最適化を行わずベクトルキャッシュを利用するだけでも，消費エネルギの大きいメインメモリへのアクセスを，消費エネルギの小さいベクトルキャッシュへのアクセスに置き換えることができ，消費エネルギが削減できます．評価結果より，最大で50 %の消費エネルギが削減可能であることがわかります．さらに，プログラム最適化を行うことにより，ループアンローリングによるメモリアクセス削減や，キャッシュブロッキングによるキャッシュヒット率の向上によってキャッシュが活用され，最大で77%の消費エネルギの削減が可能です．このように，提案するプログラム最適化戦略により，実行効率の向上だけでなく消費エネルギも大幅に削減できることができます．

以上述べてきた通り，私の研究はベクトル型スーパーコンピュータをより利用しやすくするための研究であるため，皆様の利用相談のサポートを行いながら，自分の研究にも反映していきたいと考えています．微力ですがどうぞよろしくお願いいたします．

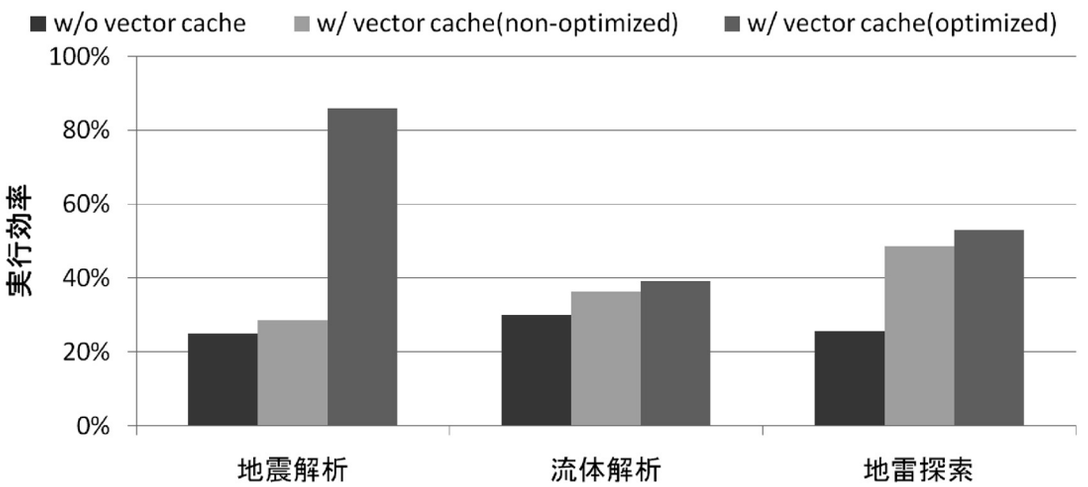

図 2 プログラム最適化による実行効率の向上

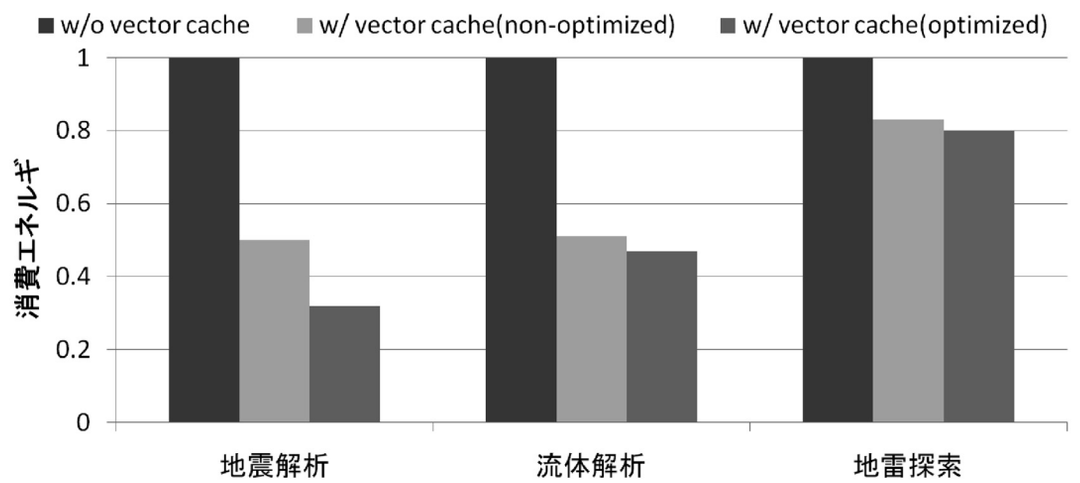

図 3 プログラム最適化による消費エネルギの削減

参考文献

[1] Hiroaki Kobayashi., “Implication of Memory Performance in Vector-Parallel and Scalar-Parallel HEC Systems”, High Performance Computing on Vector Systems 2006, Springer-Verlag, pp.22-50, 2006.

[2] Samuel Williams, Andrew Waterman, and David Patterson., “Roofline: an Insightful Visual Performance Model for Multicore Architecture”, Communications of the ACM, Vol. 52, No. 4, pp.65-76, 2009.

[3] Keisuke Ariyoshi, Toru Matsuzawa, and Akira Hasegawa., “The key frictional parameters controlling spatial variations in the speed of postseismic-slip propagation on a subductionplate boundary”, In: Earth and Planetary Science Letters, Vol. 256, pp.136-146, 2007.

[4] Himeno benchmark, http://w3cic.riken.go.jp/HPC/HimenoBMT

[5] Takeo Kobayashi, Motoyuki Sato., “FDTD simulation on array antenna SAR-GPR for landmine detection”, In: Proceedings of SSR2003, pp.279-283, 2003.